

WHAT IS CLAIMED IS:

1. A method, comprising:
 - determining a set of programming statements associated with a multithreaded
5 network processing element, the network processing element having a local memory;
 - arranging for a first portion of the local memory to be allocated to a first thread
context in accordance with a programming statement that is associated with a first thread
and symbolically references a buffer name; and
 - arranging for a second portion of the local memory to be allocated to a second
10 thread context in accordance with a programming statement that is associated with a
second thread and symbolically references the buffer name.
2. The method of claim 1, wherein said determining comprises retrieving the set
of programming statements from a storage device.
15
3. The method of claim 1, wherein said arranging comprises translating the
programming statements into code.
4. The method of claim 1, further comprising:
 - arranging for information associated with the first thread context to be stored in
20 the first portion of the local memory; and
 - arranging for information associated with the second thread context to be stored in
the second portion of the local memory.

5. The method of claim 1, further comprising:
freeing the second portion of the local memory in accordance with another programming statement that symbolically references the buffer name.
- 5
6. The method of claim 1, wherein the symbolic reference to the buffer name may be passed in at least one of: (i) a function, and (ii) a macro.
7. The method of claim 1, further comprising:
10 translating the set of programming statements into code; and
providing the code.
8. The method of claim 7, wherein the provided code is associated with at least one of: (i) assembly language, and (ii) microcode.
- 15
9. The method of claim 1, wherein the local memory comprises at least one of: (i) memory at the network processing element, (ii) hardware registers at the network processing element, and (iii) a local cache.
- 20
10. The method of claim 1, wherein the network processing element is a reduced instruction set computer microengine in a network device.
- 25
11. The method of claim 10, wherein the network device is associated with at least one of: (i) information packet header parsing, (ii) exception packet identification, (iii) information packet receipt, (iv) information packet transformation, and (v) information packet transmission.

12. The method of claim 10, wherein the network device is associated with at least one of: (i) Internet protocol information packets, (ii) Ethernet information packets, (iii) asynchronous transfer mode protocol, (iv) a local area network, (v) a wide area network, (vi) a network processor, (vii) a switch, and (viii) a router.

13. The method of claim 12, wherein the set of programming statements includes at least one of: (i) an allocate buffer instruction, (ii) a bind buffer address instruction, (iii) an activate buffer instruction, (iv) a deactivate buffer instruction, and (v) a free buffer instruction.

14. An article, comprising:

a storage medium having stored thereon instructions that when executed by a machine result in the following:

determining a set of programming statements associated with a multithreaded network processing element, the network processing element having a local memory,

arranging for a first portion of the local memory to be allocated to a thread context in accordance with a programming statement that is associated with a first thread and symbolically references a buffer name, and

arranging for a second portion of the local memory to be allocated to a second thread context in accordance with a programming statement that is associated with a second thread and symbolically references the buffer name.

15. The article of claim 14, wherein execution of the instructions further results in:

translating the set of programming statements into code.

16. The article of claim 15, wherein execution of the instructions further results
in:

providing the code.

5

17. An article, comprising:

a storage medium having stored thereon a set of programming statements adapted
to be translated into code, said programming statements including:

10 a programming statement associated with a first thread that uses a buffer
name to symbolically reference information that a first thread context will store in
local memory at a multithreaded network processing element; and

a programming statement associated with a second thread that uses the
buffer name to symbolically reference information that a second thread context
will store in the local memory at the network processing element.

15

18. The article of claim 17, where said programming statements further include:

a programming statement that uses the buffer name to symbolically reference
information that the second thread context will no longer store in the local memory at the
network processing element.

20

19. A method, comprising:

defining a programming statement that is associated with a first thread and uses a
buffer name to symbolically reference information that a first thread context will store in
local memory at a multithreaded network processing element; and

defining a programming statement that is associated with a second thread and uses the buffer name to symbolically reference information that a second thread context will store in the local memory at the network processing element.

5 20. The method of claim 19, further comprising:
arranging for the programming statements to be translated into code.

21. A system, comprising:
a processor; and
10 a hard disk drive having stored therein instructions that when executed by a machine result in the following:
translating C programming language instructions into code, and
translating an additional programming statement into code, the additional programming statement using a buffer name to symbolically reference
15 information that a thread context will store in local memory at a multithreaded network processing element

22. The system of claim 21, further comprising:
an interface to facilitate a transfer of the code from the system to the network
20 processing element.